DLSS 5: Real-Time Generative Rendering

NVIDIA

Figure 1 | Rendered input and DLSS 5 output at matched settings with enlarged crops. Professional artists use DLSS 5 controls to produce realistic material and illumination that are difficult to achieve efficiently through conventional rendering. The output remains faithful to the authored scene.

Abstract:

We present DLSS 5, a real-time generative rendering stage for interactive graphics that complements conventional rendering with appearance priors learned from real-world visual data. Prior DLSS technologies use learned reconstruction to efficiently approximate outputs that would otherwise require a higher rendering budget. This improves rendering efficiency and image quality, but realism remains bounded by what the renderer can represent and simulate. In practice, real-time compute and memory budgets limit both authored scene representations and rendering algorithms, preventing them from fully capturing real-world appearance. Foundational generative models offer a complementary source of realism. They learn broad knowledge of real-world appearance and can synthesize complex visual effects that are difficult to represent faithfully with authored assets and real-time simulation, such as subsurface scattering in skin and light scattering through foliage. However, directly applying these models to interactive rendering presents two fundamental challenges. First, general-purpose generative models do not reliably preserve developer-authored scene content and structure, even when their outputs appear visually plausible. Second, their inference cost exceeds the strict frame budgets of interactive rendering, particularly at 4K output resolution.

DLSS 5 is designed to harness learned appearance priors while faithfully preserving developer-authored content and design intent. To achieve this within an interactive frame budget, we introduce *3D-guided neural rendering*, a renderer-grounded generative approach. The system uses a one-step pixel-space diffusion model designed for high-resolution, real-time rendering. At inference, the model is conditioned on the current rendered frame, engine motion vectors, carried temporal state, and artistic-direction values. During training, consistency supervision from renderer-derived scene attributes further grounds generation to remain faithful to the authored scene. The result-

ing inference process is causal and deterministic, is trained for frame-to-frame temporal stability, and operates under a strict per-frame compute budget, enabling real-time rendering at up to 4K resolution.

DLSS 5 is the first DLSS technology to generate the final displayed appearance rather than reconstruct a higher-cost reference output from the conventional renderer. To the best of our knowledge, it is also the first productized generative rendering model to operate in real time. DLSS 5 runs locally as a rendering stage within existing game pipelines on GeForce RTX 50 Series GPUs. Generative rendering does not replace physically based rendering or artistic authorship; it complements them by supplying learned real-world appearance knowledge that additional samples and explicitly authored scene representations alone cannot efficiently provide.

1. Introduction

Real-time rendering builds every image from an explicit description of the scene. The application supplies geometry, materials, lights, simulation state, and cameras; the renderer turns that state into a frame with a prescribed algorithm. Because every pixel follows from authored state, the result is consistent, controllable, and fast enough to respond to the player.

Foundational visual generative models learn a different route to realistic images. Trained on photographic data, they capture how the real world looks — how skin, hair, foliage, and everyday materials appear under real light — as statistical properties of photographs rather than as physical models (Karras et al., 2022, 2024; Esser et al., 2024; Imagen 3 Team, 2024). DLSS 5 combines the two: it conditions a real-time generative model on the renderer's own output, so that learned appearance is applied to the scene the developers authored.

This introduction motivates that combination (Sec. 1.1), then states what DLSS 5 is, the principle it is built on, and the four requirements that structure the rest of the report (Sec. 1.2), and closes with a simplified conceptual formulation of the generation problem (Sec. 1.3).

1.1. Motivation

The cost of first-principles realism. Real-time image quality has advanced through programmable shading, general-purpose GPU computing, ray tracing, and neural reconstruction, and each step has widened what fits inside an interactive budget. But the closer rendering gets to the real world, the more every remaining step costs: more computation, more memory bandwidth, and more artist time for materials, assets, and per-scene lighting. To sustain 60 fps, the application has a budget of roughly 16.7 ms per frame, and most of it is already spent on simulation, animation, and base rendering. Photorealism is, in practice, limited less by knowledge of how to achieve it than by the scale of computation, memory, and asset production required — some light transport, such as scattering within skin, is impractical to simulate at any budget, while most content is simply authored below achievable fidelity to fit platform budgets. Section 5.2 develops this argument and the complementary, learned-appearance axis DLSS 5 adds.

Reconstruction is bounded by its target. Every prior DLSS technology reconstructs. Super Resolution, Frame Generation, and Ray Reconstruction each estimate what the renderer would have produced with more pixels, more frames, or more rays (NVIDIA Corporation, 2025b), and they deliver that estimate far more cheaply than rendering it directly. The renderer defines their target, and that target is also their ceiling: a reconstruction cannot look better than the output it reconstructs.

© 2026 NVIDIA. All rights reserved.

Appearance learned from data. The knowledge that rendering budgets cannot provide can instead be learned from photographic data by visual generative models. However, such models usually do not meet the requirements of interactive rendering: inference is typically too slow, outputs may vary from run to run, and generated content and structure may not fully align with the input CG frame. DLSS 5 brings this learned knowledge to interactive rendering through a model designed for real-time performance, deterministic inference, and strict alignment with the rendered scene.

1.2. What DLSS 5 is

DLSS 5 is a generative rendering stage — an approach we call *3D-guided neural rendering*. It runs after the renderer, inside the frame budget, and synthesizes the final displayed image from the rendered frame, for rasterized, path-traced, and hybrid pipelines alike. The model is a compact one-step pixel-space diffusion model learned from real-world prior (Sec. 2). It changes how materials and illumination look while staying conditioned on the frame the renderer produced (see Fig. 1). And because it is conditioned on the rendered frame, better rendering upstream directly improves its output: DLSS 5 composes with rasterization, ray tracing, and path tracing rather than competing with them.

The rendered frame is the primary conditioning signal: dense and pixel-aligned, it pins down what is in the scene, while artistic direction determines how far appearance may move. Neither half is enough on its own. An offline generative model can produce a beautiful image of the wrong scene; a pass-through preserves the scene without improving it. DLSS 5 targets perceived realism, scene fidelity, and artistic direction at the same time — and it deliberately targets *realism*, not restyling (Sec. 5).

The design is specified by four requirements:

- **Artistic intent preservation.** The output preserves what the artists authored: geometric structure, object and character identity, material identity, composition, and the organization of light and shadow. Nothing enforces this at runtime — the supervision that encourages it acts during training, and every generated change carries some risk of drifting from the authored scene. Section 2.1 describes the supervision, and Section 4 measures the preservation.
- **Streaming.** Inference is causal: one displayed output per rendered input, with no dependence on future frames. Offline video generators usually buy temporal coherence by processing windows of frames together; an interactive renderer cannot wait for frames that have not happened yet (Sec. 2.5).
- **Budget.** One denoising evaluation per rendered frame, within the compute and memory the application can spare. Sections 2–2.6 describe how conditioning, learning photo-realistic priors, one-step inference, and a hardware-oriented design close the gap.
- **Artistic direction.** Appearance-model selection, strength values, chroma adjustment, and spatial and semantic masking decide which transformation is applied, how strongly, and where; given the same rendered sequence and settings, the output sequence is unchanged (Sec. 3).

The first three requirements decide whether generation can run inside an interactive rendering loop at all; artistic direction decides whether the result is usable in an authored title.

1.3. Problem Formulation

This subsection gives a simplified, conceptual account of the generation problem. For clarity it considers a single frame in isolation and omits the temporal and artistic-direction conditioning of

the deployed system (Secs. 2 and 3); it is not a statement of the training procedure.

A conditional generative image model represents a distribution $p_\theta(x \mid c)$ over images x given a conditioning signal c, such as a text prompt. Training shapes p_θ so that samples are plausible natural images consistent with c; inference *draws* from this distribution, and repeated draws are expected to differ.

DLSS 5 is asked to do something narrower. Its conditioning signal is the rendered frame r, and the output should not be an arbitrary plausible image consistent with r: it should be a photographic rendition of exactly the scene that r depicts. Let p_{photo} denote a distribution of photographic appearance, learned statistically from photographic data, and let $\mathcal{C}(r)$ denote the set of images that preserve the renderer-authored attributes of r: geometric structure, object and character identity, material identity, composition, and light-and-shadow organization. DLSS 5 evaluates a deterministic map $x = f_\theta(r)$, trained so that, informally,

$$f_\theta(r) \;\approx\; \arg\max_{x \in \mathcal{C}(r)} \; p_{\text{photo}}(x) \tag{1}$$

— among images that keep the authored content, prefer the one that looks most like a photograph. The intended effect can be stated directly: the displayed output should be substantially more probable under the photographic-appearance distribution than the rendered input it replaces,

$$p_{\text{photo}}(f_\theta(r)) \;\gg\; p_{\text{photo}}(r)\,, \tag{2}$$

while remaining within $\mathcal{C}(r)$. While a text-conditioned generator spreads probability mass across the many images compatible with its prompt, DLSS 5 concentrates on the content-preserving neighborhood of its rendered input and moves the image toward the photographic-appearance distribution.

Both ingredients are trained tendencies rather than runtime guarantees: membership in $\mathcal{C}(r)$ is encouraged by training-time consistency supervision (Sec. 2.1), and p_{photo} is the photo-realistic appearance piror learned from real-world data (Sec. 2). The deployed system extends this single-frame picture in two ways described later in the report: the map runs causally over a frame stream with carried temporal state (Sec. 2.5), and the residual ambiguity — many photographic appearances are compatible with one rendered frame — is resolved by the artistic direction of Section 3 rather than by stochastic sampling.

2. System Overview

DLSS 5 is a renderer-conditioned streaming video model that processes each frame causally as it arrives. This section separates five aspects of the system: grounded conditioning (Sec. 2.1), construction of the task-specific student (Sec. 2.2 and 2.3), its pixel-space formulation (Sec. 2.4), causal temporal-state streaming (Sec. 2.5), and hardware-oriented network design (Sec. 2.6).

2.1. Grounded Conditioning

The inference interface uses the engine-rendered RGB image as a dense, registered observation of visible scene appearance. It provides dense, pixel-aligned evidence for object support, occlusion boundaries, composition, and local material properties; engine motion vectors separately provide temporal correspondence. Existing image generative models commonly rely on text embeddings, exemplar images, or spatial control fields such as depth, edges, segmentation, and pose (Esser et al., 2024; Imagen 3 Team, 2024; Black Forest Labs et al., 2025; Wu et al., 2025; Li et al., 2024). These conditions are effective for general-purpose generation and editing, but they do not

uniquely determine the object identities, materials, visibility relationships, lighting decisions, and pixel-aligned detail contained in an engine-rendered frame. DLSS 5 is therefore conditioned on the rendered frame itself.

Native image-editing models can also modify images in a spatial and dense manner (Black Forest Labs et al., 2025; Wu et al., 2025). However, their per-pixel fidelity is not yet sufficient for production rendering: small objects, thin structures, distant characters, and embedded text may be altered, displaced, or removed during editing. Some of these errors arise because model capacity is distributed across a broad range of tasks; in latent-space systems, additional error can come from the lossy autoencoder round trip through which the edited image is re-synthesized (Sec. 2.4). Interactive rendering allows little tolerance for such changes: the displayed stream must preserve authored content in every frame.

Conditioning on rendered RGB alone, however, provides a very strict constraint on the output that introduces challenges in achieving high realism. RGB entangles scene content with the appearance already produced by the renderer. If the model follows the input too literally, it tends to reproduce the same edges, colors, textures, and baked shading, leaving too little freedom to improve the final appearance. If the model is given too much freedom, it can drift from the authored content.

We resolve this challenge during training rather than at the runtime interface. During training, we use renderer-derived albedo, surface normal, and lighting information to provide attribute-specific consistency supervision. Albedo provides illumination-independent chromatic information, surface normal provides local orientation information, and the lighting-derived signals supervise the authored illumination structure. These signals are used only during training and are not inputs to the shipped model. Artistic intent that is not already expressed in the rendered frame enters through the artistic-direction control of Section 3.

These supervision targets constrain selected renderer-authored scene attributes, but they do not define a target photographic appearance. Photographic appearance is instead learned statistically from photographic data, while renderer-derived supervision discourages changes that would conflict with the authored scene. Within this supervision envelope, the model is free to change what the authored attributes do not pin down: fine surface shading, the way materials appear under light, and the overall tone of ambient and indirect illumination.

The grounding is essential for preserving object identity and maintaining semantic alignment with the rendered frame. For example, albedo consistency helps preserve the intrinsic chroma of an object while allowing the model to synthesize more realistic local illumination. The lighting supervision anchors the direct-lighting structure and helps preserve the intended placement and orientation of shadows.

2.2. Learning Photo-realistic Priors

Foundational generative models trained at scale on photographic data encode a rich prior over real-world appearance, including the statistical relationships among materials, illumination, surface geometry, and scene context as they appear in photographs. These learned appearance statistics capture effects that would be prohibitively expensive to reproduce through first-principles simulation at production scale (Sec. 5.2). However, such models are unsuitable as real-time rendering components. Because they are usually designed to accept diverse forms of conditioning and synthesize open-ended content from scratch, they require substantial model capacity and typically rely on iterative inference, placing them far outside an interactive frame budget.

Real-time rendering enhancement is a substantially narrower problem. The rendered frame already

specifies the scene content, geometry, composition, and motion; the model only needs to learn how to enhance its appearance. DLSS 5 therefore uses a dedicated network specialized for this task, focusing its capacity on the real-world appearance priors relevant to rendered frames. This specialization trades open-ended generative capability for efficiency: the resulting model is substantially smaller and faster, while retaining rich knowledge of real-world materials, lighting, and appearance.

Figure 1 illustrates this behavior on representative rendered frames. It shows the rendered input and the corresponding DLSS 5 output, together with enlarged crops highlighting the most visible changes. The enhanced outputs remain aligned with the rendered content while exhibiting richer material and illumination appearance consistent with the learned prior. Systematic comparisons and content preservation measurements are presented in Sec. 4.

2.3. One Step Diffusion

DLSS 5 employs a single-step diffusion inference rather than iterative stochastic sampling. Each enhanced frame is produced in a single deterministic inference step; given the same rendered sequence and configuration, the system produces the same output sequence. Although additional sampling steps can improve distribution modeling in general-purpose diffusion models, they do not necessarily yield substantially better perceptual quality for this strongly conditioned enhancement task. As demonstrated in Sec. 4, DLSS 5 achieves high perceptual quality within a real-time frame budget.

2.4. Pixel Space Diffusion

Most large diffusion models operate in the latent space of a variational autoencoder (VAE): images are encoded into a compact latent representation, denoised there, and decoded back to pixels (Rombach et al., 2022). This reduces computation, but the encode–decode round trip is lossy. Fine structure is re-synthesized by the decoder rather than carried through, and small latent-space deviations surface as shifted edges, altered textures, and re-drawn detail. A VAE is therefore not accurate enough to guarantee the art, narrative, and semantic preservation that a production-grade real-time generator requires: encoding and decoding introduce drift in exactly the attributes DLSS 5 must preserve.

DLSS 5 instead performs diffusion directly in pixel space. The choice is dictated by the preservation requirements above rather than by precedent; pixel-space generation at scale has likewise been favored in recent work (Hoogeboom et al., 2023; Yu et al., 2026). No learned autoencoder appears in the inference path: the model consumes rendered RGB and produces displayed RGB, and the network's internal feature maps (Sec. 2.6) serve as intermediate computation rather than as a lossy bottleneck through which the image must be reconstructed.

Generating in the same space in which the renderer produced the image is what allows DLSS 5 to leave the renderer's structure untouched. In practice, DLSS 5 does not change geometry or structural edges, and its preservation extends to the finest level of the input signal: even the individual pixel aliasing of the rendered input survives generation.

2.5. Real-time Frame Streaming

DLSS 5 targets real-time interactive rendering, in which the model operates on a continuous and unbounded stream of frames. Because inference is performed on consumer GPUs and each output must be presented with minimal delay, the system must operate under strict end-to-end latency and peak-memory limits. In addition, the model must nevertheless preserve semantic fidelity to the

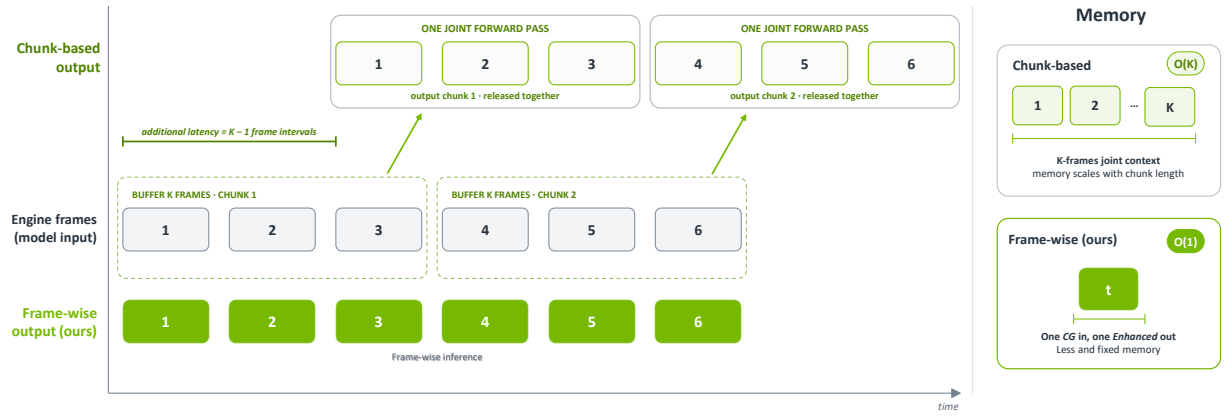


Figure 2 | Latency and memory comparison between chunk-based and frame-wise streaming inference. Chunk-based models wait for a complete K-frame chunk and process its frames jointly, introducing additional latency and a working set that scales with K. In contrast, our frame-wise method processes each renderer frame causally with less memory consumption, enabling lower latency with higher inference resolutions. $K = 3$ is shown for illustration.

rendered scene while maintaining temporal stability over extended gameplay sessions.

Chunk-based architectures are widely used to generate temporally coherent videos (Wan et al., 2025; Hunyuan, 2025; NVIDIA, 2026). These architectures process multiple frames jointly, typically using temporal attention to exchange information within a finite chunk. Although this strategy is effective for improving temporal stability, it is poorly suited to high-resolution real-time interactive rendering for two primary reasons. First, multiple frames must be buffered and jointly processed before the corresponding outputs become available. As illustrated in Figure. 2, this buffering introduces additional latency between a user action and the corresponding displayed response, thereby reducing interactivity. Second, the computational and memory costs of joint spatiotemporal processing increase with the number of frames in each chunk. Under a fixed hardware budget, distributing these resources across multiple frames constrains the budget available to each frame and can therefore limit the maximum spatial resolution.

In contrast, DLSS 5 adopts a causal frame-streaming formulation that performs immediate inference once the current renderer RGB frame is ready, and it requires no future frames. Consequently, each enhanced frame becomes available shortly after its corresponding renderer frame is processed, while the peak inference memory remains small and constant with respect to sequence duration. Rendered RGB frames anchor the corresponding predictions to the current scene contents, preserving semantic fidelity, while the streaming model maintains temporal coherence across successive enhanced frames. This formulation enables low-latency, high-resolution enhancement, including at 4K, and is therefore well aligned with the requirements of real-time gaming.

2.6. Real-time Backbone

The implementation is constrained by per-frame latency, memory usage, output quality, and temporal stability. To achieve all of these goals, we co-optimized network topology, numerical precision, and GPU kernels, rather than treating any of these as an isolated problem. Meeting these targets requires reducing inference time by approximately three orders of magnitude relative to representative offline diffusion transformers at 4K. We combined automated architecture search with targeted ablations to identify network architectures that map efficiently to Tensor Core execution while retaining

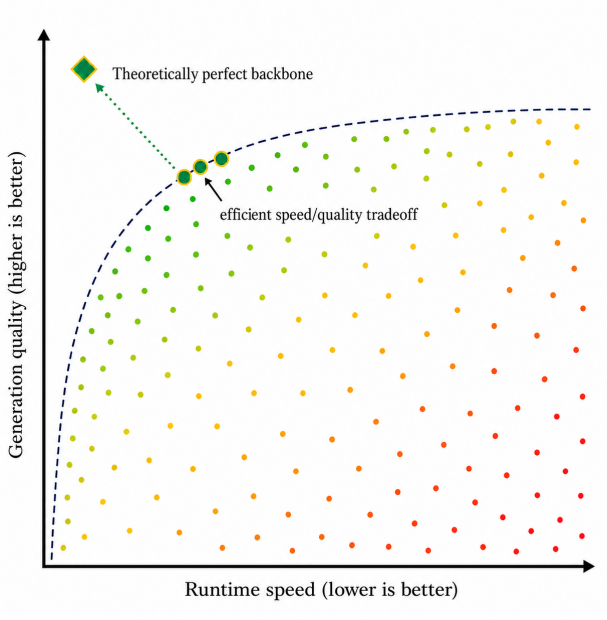


Figure 3 | DLSS 5 is optimized to achieve a favorable tradeoff between performance and quality.

sufficient model capacity.

As illustrated in Fig. 3, we evaluated a broad range of candidate architectures and selected a backbone that provides an efficient tradeoff between generation quality and runtime performance. We currently optimize the implementation for Blackwell GPUs with TMA instructions. The resulting architecture is a transformer of approximately 154 million parameters.

Architecture DLSS 5 uses a transformer backbone. Its attention layers build on the transformer architecture introduced in DLSS 4 (NVIDIA Corporation, 2025b), which provides a favorable speed-quality operating point and benefits from mature software optimization on the target hardware.

Model Optimizations The attention layers reuse implementation techniques described in our prior work (NVIDIA Corporation, 2025b) and adopt kernel-design techniques established for large language models and diffusion transformers. Tensor Memory Accelerator (TMA) operations stage data from global memory into on-chip storage (NVIDIA Corporation, 2025a).

Reduced Precision Most GEMMs and attention operations use FP8, while a small number of numerically sensitive GEMMs remain in FP16. Compared with an otherwise equivalent FP16 implementation, the FP8 configuration introduces only a small measured quality difference while providing an approximately 2× improvement in inference speed

3. Artistic Direction

DLSS 5 exposes conditioning variables selected by the developer to specify transformation strength, color response, and overall appearance. These variables enable the developer to control the artistic appearance of the transformation that is most appropriate for their game.

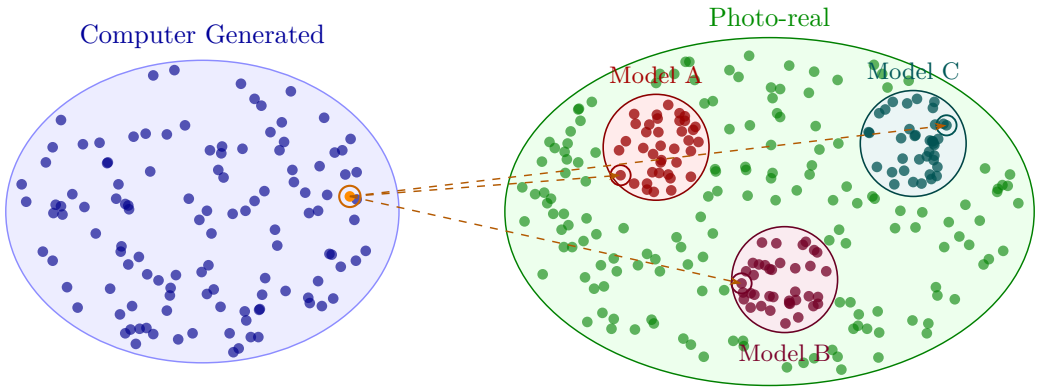


Figure 4 | An illustration of the transport between source and target distributions.

(a) CG (b) Model A (c) Model B (d) Model C

Figure 5 | A small set of curated appearance models of DLSS 5 that sample different regions of the broad photo-realistic image distribution.

3.1. Selecting an Appearance

One useful framework we find is to consider that DLSS 5 is a transformation from a distribution over computer generated frames to a target distribution (see Fig. 4). In practice, even the distribution over "photographs" varies wildly and encompasses many choices like film, digital, lens, color grading, and tonemapping. Ideally, DLSS 5 would be both large enough and fast enough to provide enough control to explore the entirety of plausible photographic mappings in a single model, but this remains an open and difficult problem. As an approximation, the current system samples this family with a small set of curated models whose learned output distributions differ in tone, color, and material appearance, such as in their treatment of vegetation, light, and skin to name a few, as shown in Figure 5. Together, these models represent a discretized sampled subset of the available photographic-appearance mappings.

As 3D-guided neural rendering matures and evolves, we expect more and more control over the diversity of target styles to emerge, allowing developers and artists to find ideal mappings that align with their game's desired genre, tone, contrast, lighting and material response.

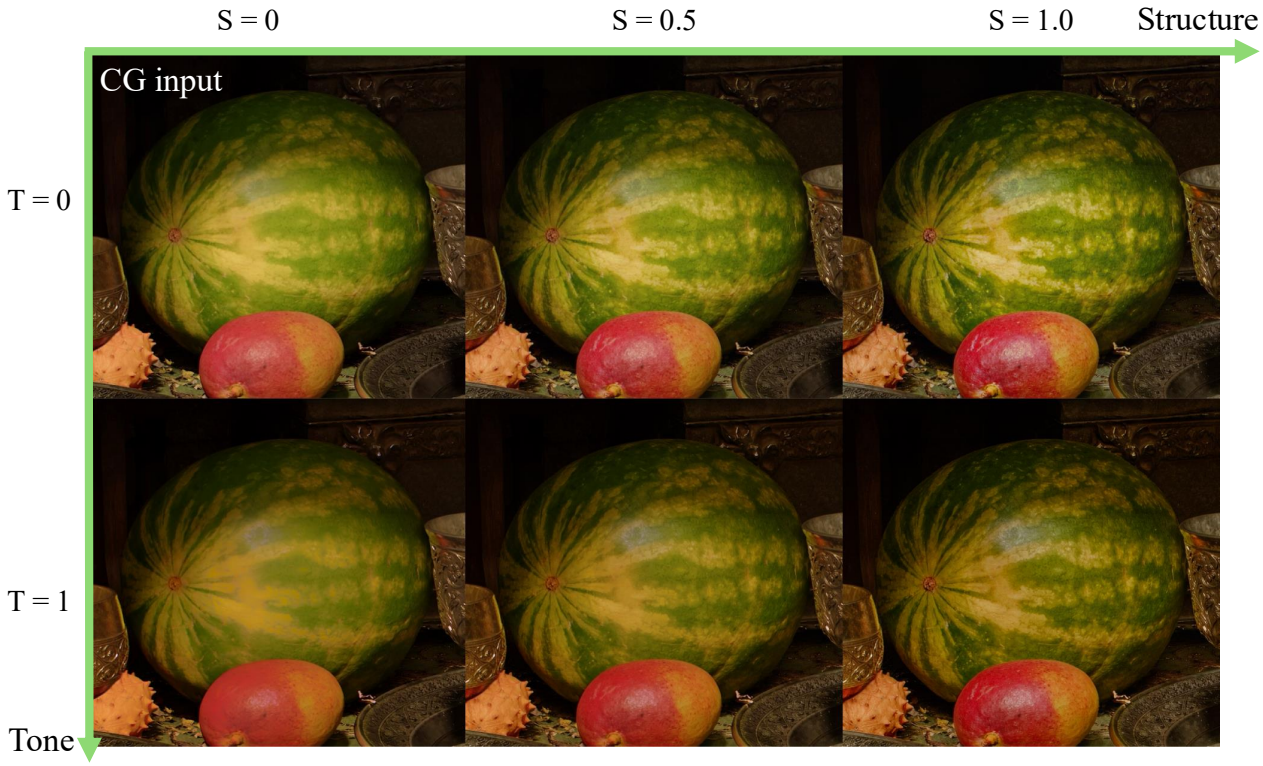


Figure 6 | **Structure and tone control:** the structure control adjusts structure detail enhancement, while the tone control matches the CG image's overall tone.

3.2. Separating Tone and Structure

In addition to output style, artists need to determine where and how much enhancement is applied. We observed that the overall enhancements produced by DLSS 5 can be intuitively categorized as (1) changes in fine detail of surfaces and adjustments to local contrast, (2) shifts in broad luminance and illumination of scene elements. To enable independent adjustment of each category of enhancement, DLSS 5 exposes *structure* and *tone* strength modulation parameters respectively. At zero strength, the system is configured to return an enhancement-free input; at unit strength, it applies the full learned transformation. As implemented, the *structure* and *tone* controls behaviorally correspond to high and low spatial frequency bands of the image content.

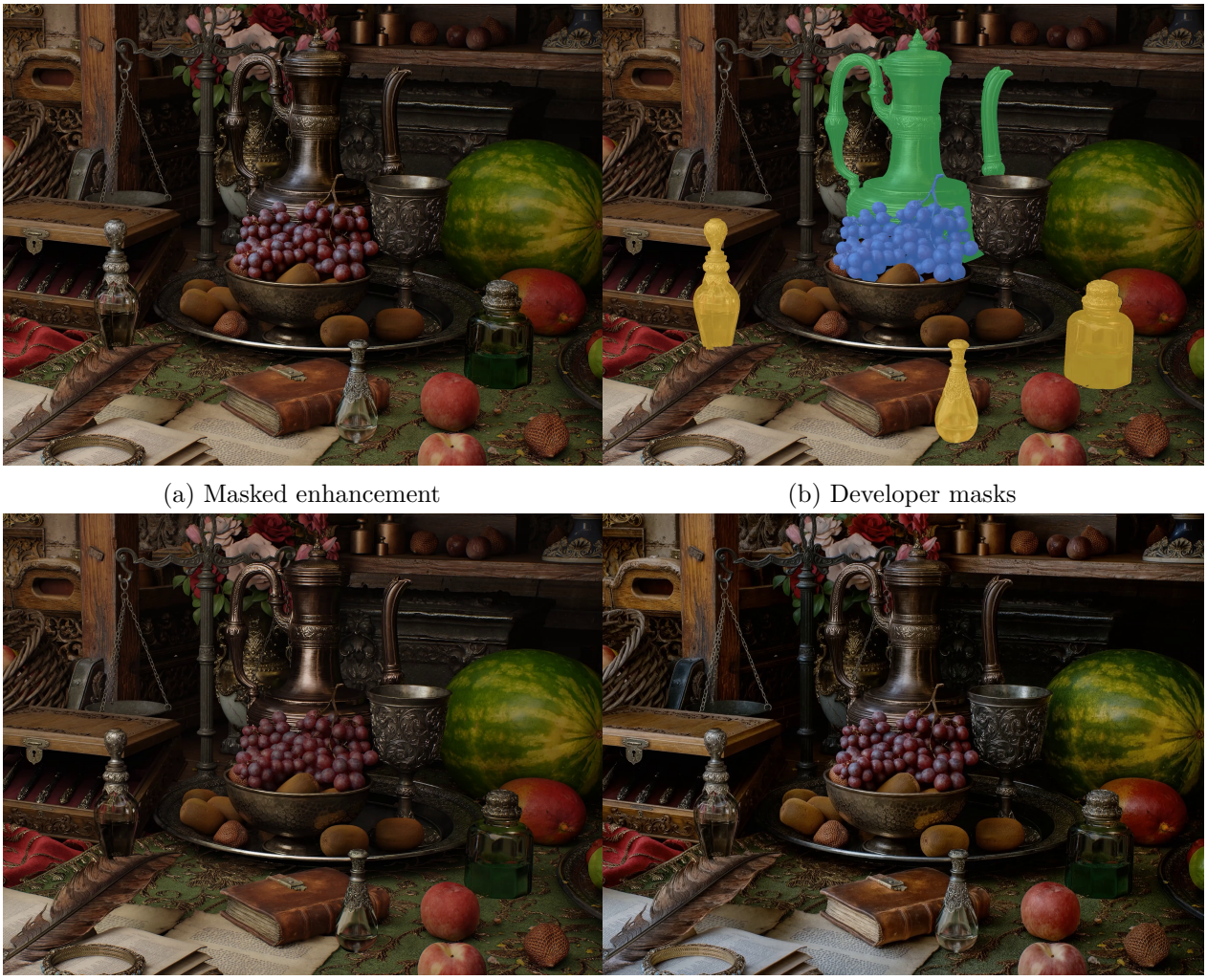
Fig. 6 illustrates the effects of the two strength coefficients on the target output.

The structure strength, tone strength, and model subset parameters are incorporated as learned conditioning inputs. This simplifies the deployed algorithm, and fuses an expensive multi-frequency decomposition operation into the network which minimizes the resulting run time at inference.

3.3. Auto and Manual Masking

In addition to strength, it is important to offer control over *where* the enhancement applies. Artists will often welcome substantial variation in grass, stone, or wood, but require precise control over characters and other hero assets. DLSS 5 therefore determines where enhancement applies through two complementary masking mechanisms: regional manual masking of the strength and tone parameters, and regional automatic masking of semantic features like skin.

Manual masking. Artists familiar with their engine can often easily specify per-asset the tone



(a) Masked enhancement (b) Developer masks

(c) CG (d) Global enhancement

Figure 7 | Manual masking applies enhancement selectively to developer-defined regions. The masked result in (a) uses the masks shown in (b) to control enhancement relative to the CG input (c), while (d) applies enhancement throughout the image.

and strength parameter per pixel and provide a 2D surface of these per-pixel scalars to DLSS 5 for masking. Like global tone and structure strength, the effect of these per-pixel masks is also encoded into the model during training and the masks are added as channel inputs at run time. Fig. 7 shows the manual masking effects of DLSS 5. In addition to the speed performance improvement, we also found that mask transitions between objects are made much smoother by the network as a result of being object and material aware.

Manual masking in this way is the most comprehensive approach to control but comes at the expense of developer time in creating and rendering the mask inputs.

Automatic masking. We found that many developers were using manual masking for common semantic-level elements such as skin. To avoid the run time and developer expense of creating a mask, we learn a semantic mask into the model for skin (see Fig. 8) This exchanges a manually generated per-pixel mask for a new global strength parameter that applies only to pixels that match

(a) CG Input (b) Enh. only on non-skin (c) Full enhance (d) Enh. only on skin

Figure 8 | Shows the CG source image and three possible outputs of DLSS 5 that vary tone and structure between 1 (enhanced) and 0 (not enhanced), depending on the automatically inferred skin-mask region. Enh. denotes to enhancement.

the "skin"[1] semantic group. This automatic semantic masking demonstrates the model's learned understanding of scene semantics. We leverage this capability to provide artists with semantic controls without requiring manually authored masks.

Automatic masking demonstrates that a learned statistical representation can support content-specific modulation through an artist-facing parameter. Future work can extend parameters like these to additional content categories such as hair, foliage, fabrics, individual materials, or potentially developer-defined semantic groups or individuals.

4. Evaluation

We evaluate DLSS 5's results based on three complementary aspects: perceived image quality and photographic realism, content and identity alignment with the renderer-authored input, and structure alignment with the rendered scene. We use the same evaluation content across methods wherever possible so that these properties can be interpreted jointly. At the end of this section, we also report the speed and VRAM of DLSS 5.

4.1. Setup

Evaluation data. Our quantitative evaluation contains 109 aligned scenes. The set includes both path-traced (PT) and non-path-traced (non-PT) rendered inputs and covers characters, environments, complex materials, and varied lighting. The main comparison reports results over the full set. We additionally report a PT/non-PT split as a robustness check in Sec. 4.3.2; the two subsets contain different scenes and therefore should not be interpreted as a controlled causal comparison of rendering quality.

Compared methods and references. We compare DLSS 5 with two offline foundation models, GPT Image 2 (OpenAI, 2026) and Gemini 3 Pro (Nano Banana Pro) (Google DeepMind, 2025). All methods operate on the same CG inputs. Since image-editing models can respond differently to the same spatial constraints, we use a fixed, model-specific prompt for each baseline to balance

[1] To our delight, we found the model tends to treat (within reasonable limits) alien and reptilian skin the same as human skin.

photorealism and content preservation.[2] Each prompt is kept fixed across all evaluation scenes.

For image-quality evaluation, we directly compare the generated outputs through the blinded user study. For alignment evaluation, the original CG frame serves as the reference: content and identity alignment is measured directly in the feature space of the RGB image, while structure alignment is evaluated through the albedo, surface-normal, and depth maps of the CG and generated images, as detailed in Sec. 4.2. We also include a matched-budget renderer-scaling reference — the same scenes rendered with 4× more rays within the same total frame time — used for qualitative comparison (Fig. 10).

4.2. Metrics

Image quality / realism. We measure image quality as perceived photographic realism using a blinded pairwise user study. All model outputs are generated at 3840×2160. For the user study, participants are shown two results from different models and are required to select the result that appears more photorealistic and less CG-rendered, considering lighting, shadows, reflections, and visible artifacts. They are explicitly instructed not to penalize content differences between the two outputs. The user study therefore measures image quality and realism only, rather than preservation of the input content or structure.

Content and identity alignment. We use DINOv2 (Oquab et al., 2023) and LPIPS (Zhang et al., 2018) to measure how closely the generated RGB output preserves the visible content and identity of the CG input. Both the CG inputs and generated outputs are at 3840×2160. DINOv2 patch distance is computed after resizing both images to 672×378 and compares spatially corresponding normalized ViT-B/14 patch features using cosine distance; lower is better. Image LPIPS is computed directly between the full-resolution 3840×2160 CG input and generated output using AlexNet LPIPS, without spatial resizing; lower is better. These metrics are used strictly as content and identity preservation metrics and are not interpreted as measures of photographic quality.

Structure alignment. To evaluate whether the generated image preserves the underlying scene structure, we compare the albedo, surface normals, and depth of the generated image with those of its corresponding CG input, all at 3840×2160. We trained a single estimator to extract these buffers from both the input and the output. These comparisons measure whether the scene attributes of the generated output remain consistent with those of the CG frame.

For *albedo*, we report SSIM (Wang et al., 2004) and PSNR at the native 3840×2160 resolution, together with LPIPS computed after resizing the predicted albedo maps to 960×540. For *surface normals*, we report mean angular error, accuracy within 11.25°, and edge F1, all at 3840×2160; the angular metrics measure local surface-orientation consistency, while edge F1 measures preservation of geometric boundaries. For *depth*, we report Spearman rank correlation, SSIM, edge F1, and edge Chamfer distance at 3840×2160, capturing relative depth ordering, spatial depth structure, and depth discontinuities. Metric directions are indicated in Table 1.

[2] For Gemini 3 Pro, which already preserves identity and scene layout well under a simple semantic instruction, we use `"Generate the live-action film version of this image."` For GPT Image 2, which benefits from more explicit spatial constraints to prevent reframing and object displacement, we use `"Re-render this exact scene as a convincing real-world photograph. Keep the composition, camera, perspective, silhouettes, major edges, and four corners registered to the input; do not crop, zoom, or reframe. Freely improve local lighting, materials, skin, texture, shadows, and reflections so the result no longer looks computer-generated."` These prompts are carefully chosen to achieve a balance between content preservation and image quality.

4.3. Evaluation Results

4.3.1. Main Results

Quantitative results. Table 1 reports the alignment results over all 109 scenes. We interpret the first two rows as *content and identity alignment*. DLSS 5 obtains a DINOv2 patch distance of 0.0364, compared with 0.1405 for GPT Image 2 and 0.1223 for Gemini 3 Pro, and an image LPIPS of 0.0618, compared with 0.3386 and 0.3112. The substantially smaller distances show that DLSS 5 preserves the visible content and identity of the rendered input much more closely than the offline foundation models. Importantly, these metrics are not used to claim higher image quality.

The remaining rows measure *structure alignment* through albedo, normal, and depth. DLSS 5 is best on every reported metric: it achieves an albedo SSIM of 0.888, a normal angular error of 12.15° with 70.5% accuracy within 11.25°, and a depth Spearman correlation of 0.933 together with a 3.16-pixel edge Chamfer distance. The consistent gains across material appearance, surface orientation, depth ordering, and geometric boundaries indicate substantially stronger preservation of the renderer-defined scene structure.

Table 1 | Quantitative results for content/identity and structure alignment over 109 aligned scenes. DINOv2 and image LPIPS compare each generated RGB output with the corresponding CG input. Structure metrics compare the albedo, normal, and depth of the generated output with those of the CG input. Best results are shown in bold.

Family	Metric	DLSS 5	GPT Image 2	Gemini 3 Pro
Content/identity	DINOv2 patch distance ↓	**0.0364**	0.1405	0.1223
Content/identity	Image LPIPS ↓	**0.0618**	0.3386	0.3112
Albedo	SSIM ↑	**0.888**	0.787	0.804
Albedo	LPIPS ↓	**0.121**	0.204	0.199
Albedo	PSNR ↑	**24.79 dB**	20.98 dB	21.17 dB
Normal	Mean angular error ↓	**12.15°**	18.27°	17.84°
Normal	Acc@11.25° ↑	**70.5%**	57.7%	58.4%
Normal	Edge F1 ↑	**0.935**	0.808	0.848
Depth	Spearman ρ ↑	**0.933**	0.919	0.898
Depth	SSIM ↑	**0.963**	0.919	0.921
Depth	Edge F1 ↑	**0.820**	0.756	0.782
Depth	Edge Chamfer ↓	**3.16 px**	4.98 px	3.95 px

User study. We conduct a blinded pairwise user study to evaluate perceived photographic realism. In each question of the user study, participants are shown outputs from two models applied to the same CG input. The two outputs are presented anonymously, and participants can toggle between them for direct comparison. The CG input itself is not shown during evaluation.

Participants are asked to select the result that looks more photorealistic and less CG-rendered, considering factors such as lighting, shadows, reflections, and visible visual artifacts. Importantly, participants are explicitly instructed that the contents of the two outputs may not fully align and that such differences should not affect their image-quality judgment. This instruction is intended to separate perceived realism from the alignment measurements reported in Table 1.

The study yielded 794 usable pairwise judgments after excluding 21 judgments affected by HUD-

CG Input | Gemini 3 Pro | GPT Image 2 | DLSS 5

Figure 9 | Qualitative comparison of DLSS 5 and offline foundation models on representative gameplay scenes. DLSS 5 improves photographic appearance while remaining closely anchored to the rendered content. The examples include both the full CG input and enlarged crops highlighting characters, materials, lighting response, and fine scene structure. Please zoom in to view the details.

related bias. We report pairwise preference separately for comparisons against GPT Image 2 and Gemini 3 Pro, and further break the results down into general content and character-focused content. A win contributes one point and a tie contributes half a point; the preference score is the fraction of available points obtained by each method.

As shown in Table 2, DLSS 5 is effectively tied with GPT Image 2 overall (50.63% vs. 49.37%), while being preferred substantially more often on character-focused content (57.74% vs. 42.26%). Against Gemini, DLSS 5 is preferred both overall (62.00% vs. 38.00%) and particularly on character-focused content (66.91% vs. 33.09%). These user-study results should be interpreted only as image-quality results. They show that DLSS 5 achieves photographic realism comparable to GPT Image 2 overall and higher than Gemini overall, with particularly strong preference on character-focused content.

Table 2 | Blinded pairwise photographic-realism user study for DLSS 5. The score assigns one point to a win and half a point to a tie. "W–L–T" gives wins, losses, and ties for DLSS 5. The study contains 794 usable pairwise judgments after excluding 21 HUD-biased judgments.

Comparison	Subset	Judgments	DLSS 5 W–L–T	Preference
vs. GPT Image 2	Overall	394	169–164–61	50.63% / 49.37%
vs. GPT Image 2	Character	265	136–95–34	57.74% / 42.26%
vs. Gemini 3 Pro	Overall	400	218–122–60	62.00% / 38.00%
vs. Gemini 3 Pro	Character	272	169–77–26	66.91% / 33.09%

Qualitative comparison. Figure 9 complements the quantitative results with side-by-side comparisons on representative scenes. The examples are selected to show both the realism uplift and preservation of character identity, scene layout, materials, lighting structure, and fine geometry. We include both lower- and higher-quality rendered inputs in the same figure to illustrate how the behavior changes with the quality of the source rendering.

Generation versus scaling the renderer. A natural question is what the same frame time buys when spent on the renderer instead. Figure 10 compares DLSS 5 with the same scenes rendered using 4× more rays **under a matched total frame time**. Additional rays reduce sampling error in the illumination the renderer already computes, while DLSS 5's generation changes how materials and illumination appear. On these visual examples, DLSS 5 generates vivid contact shadow for the chair, realistic texture details on the ground, and ambient shadow and lighting on the leaves, while still maintaining the content and structure alignment with the input CG frame. Overall, DLSS 5 achieves a much better photo-realism than using additional ray tracing at a matching computation budget.

Overall analysis. Viewing all the results together, we can summarize that DLSS 5 does not obtain its realism by sacrificing fidelity to the rendered scene. DLSS 5 reaches realism comparable to GPT Image 2 overall and exceeds Gemini 3 Pro, while simultaneously preserving the input content, identity, and scene structure substantially better than both offline foundation models. This combination of realism and alignment is the central advantage of our model.

4.3.2. Effect of Input Quality

To test whether the main alignment result depends on the rendering regime, we split the evaluation set into 20 path-traced (PT) and 89 non-path-traced (non-PT) scenes and recompute a compact set of representative metrics. Table 3 shows the same ordering in both subsets: DLSS 5 has the lowest DINOv2 and image-LPIPS distances, the highest albedo SSIM, the lowest normal angular error, and the highest depth Spearman correlation. The advantage is therefore not driven by only one rendering regime. For DLSS 5, the PT subset is also numerically closer to the input than the non-PT subset on all five reported metrics (e.g., image LPIPS 0.0478 vs. 0.0649 and normal angular error 9.63° vs. 12.71°). However, because the PT and non-PT subsets contain different scenes, this cross-subset difference is descriptive only. We therefore use the split as a robustness check and do not interpret the numerical gap between the two subsets as a controlled estimate of the causal effect of path tracing or input quality.

CG Input 4× Ray Tracing DLSS 5

Figure 10 | Comparison of DLSS 5 and 4x ray tracing, which are under the same frame-time budget. Additional rays refine the illumination the renderer already computes, while DLSS 5 changes material and illumination appearance toward the photographic prior. Please zoom in to view the details.

Table 3 | Robustness across path-traced (PT) and non-path-traced (non-PT) inputs. Results are averaged over 20 PT and 89 non-PT scenes. Because the two subsets contain different scenes, PT–non-PT differences are descriptive rather than causal.

Metric	Input regime	DLSS 5	GPT Image 2	Gemini
DINOv2 patch ↓	PT	**0.0302**	0.1082	0.1237
	Non-PT	**0.0378**	0.1477	0.1220
Image LPIPS ↓	PT	**0.0478**	0.3141	0.3718
	Non-PT	**0.0649**	0.3440	0.2976
Albedo SSIM ↑	PT	**0.915**	0.756	0.787
	Non-PT	**0.882**	0.794	0.808
Normal angle ↓	PT	**9.63°**	14.83°	15.48°
	Non-PT	**12.71°**	19.04°	18.37°
Depth Spearman ρ ↑	PT	**0.974**	0.958	0.943
	Non-PT	**0.924**	0.910	0.887

4.3.3. Speed and memory.

All DLSS 5's results come at a tiny fraction of the baselines' cost. DLSS 5 runs as a rendering stage: model execution takes approximately 8 ms per 3840×2160 resolution frame on a GeForce RTX 5090 and uses 731 MB of peak GPU memory, while sharing the GPU with the running game. The offline baselines GPT Image 2 and Gemini 3 Pro are large hosted foundation models. Their exact sizes

and serving configurations are not public, but models of this class are orders of magnitude larger than DLSS 5's backbone (Sec. 2.6), run on datacenter systems rather than consumer GPUs, and take seconds or longer to produce a single image — orders of magnitude more time than DLSS 5's per-frame budget. DLSS 5 therefore achieves perceived realism comparable to the strongest offline baseline, and substantially stronger content and structure preservation than both, at interactive rates on a single consumer GPU.

5. Discussion and Limitations

5.1. Perceived Realism and Physically Based Ground Truth

A natural question is how the results in this report relate to physically based ground truth. For a fixed scene representation and light-transport model, the renderer-defined physical reference is the result approached as sampling converges. DLSS 5 does not attempt to reconstruct or estimate this reference in the way a conventional reconstruction method would. Instead, it produces a conditional output using statistical appearance priors learned from photographic data, while remaining constrained by the rendered input and by renderer-based supervision during training.

Our evaluation includes a matched-budget renderer baseline that spends the same frame time on 4× more rays (see section 4.3.1). When DLSS 5's output is judged more realistic than rendered output, that judgment should be interpreted only as evidence about perceived photographic realism. It does not imply greater fidelity to the renderer-defined physical reference. Such perceptual judgments can be influenced by statistical properties learned from photographs, including characteristic appearances of skin, hair, materials, and local shading. DLSS 5 may improve these appearance cues when they are weak in the rendered input, even when doing so does not make the result more physically accurate.

The renderer remains the authoritative source of scene content and the reference for physical correctness. DLSS 5 operates on displayed appearance rather than redefining that reference. Artistic direction determines the selected appearance transformation, including its strength or whether it is applied at all.

This distinction is both an interpretation of our results and a limitation of the method: DLSS 5 neither estimates nor explicitly targets the renderer-defined physical reference. Section 5.2 discusses why this formulation is useful in practice. The following subsections describe two additional structural limitations: the restriction to photorealistic rendering and the dependence on input quality.

5.2. The Practical Gap to First-Principles Photorealism

Photorealism is not fundamentally limited by a lack of physical understanding. With sufficiently accurate light transport, assets captured at real-world fidelity, and fully physically based material models, rendering can already produce imagery that appears photographic. However, in practice, production renderers can only approximate all three because of limits in computation, memory, and asset complexity.

Some effects remain difficult to simulate faithfully even with substantial compute. In skin, light enters the surface, scatters through layered tissue and fine structure, and re-emerges elsewhere; simulating this transport faithfully is beyond even generous offline budgets. Other materials are physically well understood but remain expensive to represent and evaluate under real-time compute and memory — semi-translucent media, silk, hair, fur, snow, and their interactions. More commonly, the limitation arises earlier in the pipeline: to meet the budgets of shipping platforms, developers

reduce texture resolution, geometric complexity, and material detail. As a result, even effects that are straightforward to model may be represented at lower fidelity than the rendering system could ideally support.

The practical gap to photorealism is therefore often a problem of scale (i.e., computation, memory, and asset production) rather than one of missing physical principles. DLSS 5 introduces a complementary approach. Instead of explicitly pushing every approximation closer to its physical reference, it uses statistical properties learned from photographic data to restore appearance cues that may be missing or weakened by those approximations, particularly in material response and illumination. DLSS 5 does not reproduce the omitted physical simulation, nor does it modify the authored scene. Rather, it is trained to enrich perceived material and lighting appearance while remaining conditioned on the rendered frame (Sec. 2.1) and subject to artistic control (Sec. 3).

5.3. Limitation to Photorealistic Rendering

The current objective of DLSS 5 is to produce vivid, photorealistic game imagery by generating realistic lighting, materials, and fine texture details. Therefore, the system is not designed for games whose visual identity depends on a strongly stylized aesthetic. For example, voxel-based, illustrated, and cartoon-like games intentionally use non-photorealistic visual languages. Applying the current DLSS 5 to such content may work against the art direction rather than improve it.

A more general system would allow developers to specify the desired rendering style and transform a game scene accordingly while preserving its geometry, semantics, and temporal consistency. Supporting such arbitrary stylistic direction is an important avenue for future work. The current version of DLSS 5, however, remains limited to photorealistic rendering.

5.4. Input Image Quality Constraint

DLSS 5 produces an RGB image and does not modify the engine's underlying geometry, scene state, or G-buffer contents. Its output is therefore fundamentally constrained by the information available in the rendered input.

Although DLSS 5 can enrich local appearance, it cannot reliably recover scene information that is absent, incorrect, or severely degraded in the input. Missing, aliased, or otherwise corrupted scene evidence may be preserved or amplified. In such cases, the rendered observation does not provide enough evidence to support a faithful enhancement for DLSS 5.

Low-quality scene rendering is particularly challenging: if object boundaries or structural details are weakly represented in the input, DLSS 5 cannot be expected to reconstruct their intended form reliably. The model is therefore better understood as an appearance-enhancement model rather than a mechanism for correcting arbitrary upstream rendering errors. It can improve the perceived realism of a valid and sufficiently informative input, but it cannot guarantee the reconstruction of missing geometry, object identity, or authored details. Figure 11 illustrates an example in which the aliasing and degraded detail in the window frames limit the quality of the corresponding region in the DLSS 5 output.

This behavior reflects a deliberate trade-off. Where the input evidence is defective, a generative model could invent a plausible replacement, but an invented replacement risks changing what the image depicts. DLSS 5 is trained to take the safer side of that trade: it preserves the geometric structure of the input exactly — including defects such as aliasing (Sec. 2.4) — rather than risk altering the story or meaning of the image. A visible defect can be fixed by improving the input; a changed meaning cannot be fixed downstream at all.

Low-quality CG Input DLSS 5 Result

Figure 11 | Example showing how degraded detail in a low-quality rendered input (left) limits the quality of the corresponding region in the DLSS 5 output (right).

This dependence on the rendered input also makes the model sensitive to distribution shifts between training and deployment. Inputs with substantially different shading conventions, material distributions, scene statistics, or artifact patterns may fall outside the training distribution and lead to degraded results. Improving robustness to imperfect inputs and a broader range of rendering configurations remains an important direction for future work.

6. Conclusion

DLSS 5 extends the DLSS family beyond pixel reconstruction to renderer-conditioned generative enhancement. It learns photographic appearance priors from data with a compact, one-step pixel-space streaming diffusion model. Renderer-derived consistency supervision during training encourages alignment with authored scene attributes, while causal frame-wise inference supports low latency and temporal stability at high resolution. The controls govern appearance style, enhancement strength, color response, and spatial and semantic masking; together with deterministic inference, they limit unintended modifications to authored content and make the selected enhancement behavior reproducible.

Our evaluation shows that DLSS 5 can operate within the target frame and memory budgets while producing a more photorealistic appearance, including richer material details and more expressive lighting. At the same time, it preserves scene structure and renderer-authored content, and maintains temporal stability. Because it operates downstream of the conventional renderer, DLSS 5 integrates learned generative enhancement into existing rendering workflows without assuming responsibility other functions including scene simulation and animation.

More broadly, integrating real-time generative models with conventional rendering pipelines can expand the expressive capacity of interactive graphics, and DLSS 5 firstly demonstrate the possibilies in this direction. This increased capacity introduces a corresponding need for greater predictability, directability, and fidelity to developers' intent. Future work will therefore pursue improvements in visual fidelity, generalization, efficiency, and temporal robustness, together with finer-grained spatial, semantic, stylistic, and temporal controls. These directions point toward a new paradigm for real-time graphics in which learned generative enhancement augments, rather than replaces, conventional rendering.

Contributors

DLSS 5 is the result of a multi-year collective research, engineering, and artistic effort across teams at NVIDIA. We are grateful to everyone who helped shape and bring it to life, and thank the following contributors, listed alphabetically by first name.

Abhijit Bhelande, Andrew Tao, Anjul Patney, Anton Schreiner, Aysegul Dundar, Bernardo Antoniazzi, Boon Cotter, Bryan Catanzaro, Chao Liu, Chase Telegin, Chris Perrella, David Dynerman, David Tarjan, Edward Liu, Emanuel Kozerski, Foozhan Ataiefard, Gabriele Leone, Hoanh Le, James Norton, Jarmo Lunden, Jason Mawdsley, Jerran Schmidt, John Sweeney, Karthik Vaidyanathan, Loudon Cohen, Monde Duinkharjav, Nannan Li, Pietari Kaskela, Robert Pottorff, Sam Ricker, Soa Lee, Songyin Wu, Sungbae Kim, Tero Karras, Timo Roman, Ting-Chun Wang, Wei Xiong, Yegor Swarovski, Yichen Sheng, Yuyang Zhao, Zheng Zeng, Ziad Ben Hadj Alouane

References

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. doi: 10.48550/arXiv.2506.15742.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. doi: 10.48550/arXiv.2403.03206.

Google DeepMind. Gemini 3 Pro Image (Nano Banana Pro). https://blog.google/innovation-and-ai/products/nano-banana-pro/, 2025. Accessed: 2026-08-31.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning (ICML)*, 2023.

Hunyuan. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

Imagen 3 Team. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024. doi: 10.48550/arXiv.2408.07009.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022. doi: 10.52202/068431-1926.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184, 2024.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. ControlNet++: Improving conditional controls with efficient consistency feedback. *arXiv preprint arXiv:2404.07987*, 2024. doi: 10.48550/arXiv.2404.07987.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

NVIDIA Corporation. NVIDIA RTX Blackwell GPU Architecture. https://images.nvidia.com/aem-dam/Solutions/geforce/blackwell/nvidia-rtx-blackwell-gpu-architecture.pdf, 2025a. Whitepaper.

NVIDIA Corporation. DLSS 4: Transforming Real-Time Graphics with AI. https://research.nvidia.com/labs/adlr/DLSS4/, 2025b. Technical report.

OpenAI. GPT Image 2. https://developers.openai.com/api/docs/models/gpt-image-2, 2026. Accessed: 2026-08-31.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ziyi Xia, Ze Liu, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. doi: 10.48550/arXiv.2506.18871.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

A. Citation

Please cite this work as "DLSS 5." Below is the BibTeX entry for citation:

```
@misc{dlss5,
  author       = {NVIDIA Corporation},
  title        = {{DLSS 5: Real-Time Generative Rendering}},
  howpublished = {\url{https://research.nvidia.com/labs/adlr/DLSS5/}},
  note         = {Technical Report},
  year         = {2026}
}
```